29 | 图神经网络：Pinterest是如何应用图神经网络的？

2020-12-21 王喆

深度学习推荐系统实战 进入课程 >

讲述：王喆
时长 14:50 大小 13.60M

你好，我是王喆。

互联网中到处都是图结构的数据，比如我们熟悉的社交网络，最近流行的知识图谱等等，这些数据中包含着大量的关系信息，这对推荐系统来说是非常有帮助的。

为了能更好地利用这些信息进行推荐，各大巨头可谓尝试了各种办法，比如我们之前学过的 DeepWalk、Node2Vec 这些非常实用的 Graph Embedding 方法。但是技术的发展永无止境，最近两年，GNN（Graph Nerual Netwrok，图神经网络）毫无疑问是最火热、最流行的基于图结构数据的建模方法。严格一点来说，图神经网络指的就是可以直接处理图结构数据的神经网络模型。

在诸多 GNN 的解决方案中，著名的社交电商巨头 Pinterest 对于 GraphSAGE 的实现和落地又是最为成功的，在业界的影响力也最大。所以，这节课我们就学一学 GraphSAGE 的技术细节，看一看 Pinterest 是如何利用图神经网络进行商品推荐的。

搭桥还是平推？技术途径上的抉择

在正式开始 GraphSAGE 的讲解之前，我想先给你讲一讲 DeepWalk、Node2vec 这些 Graph Embedding 方法和 GNN 之间的关系，这有助于我们理解 GNN 的原理。

我们这里简单回顾一下 DeepWalk 和 Node2vec 算法的基本流程，如下面的图 1 所示。它们在面对像图 1b 这样的图数据的时候，其实没有直接处理图结构的数据，而是走了一个取巧的方式，先把图结构数据通过随机游走采样，转换成了序列数据，然后再 用诸如 Word2vec 这类序列数据 Embedding 的方法生成最终的 Graph Embedding。

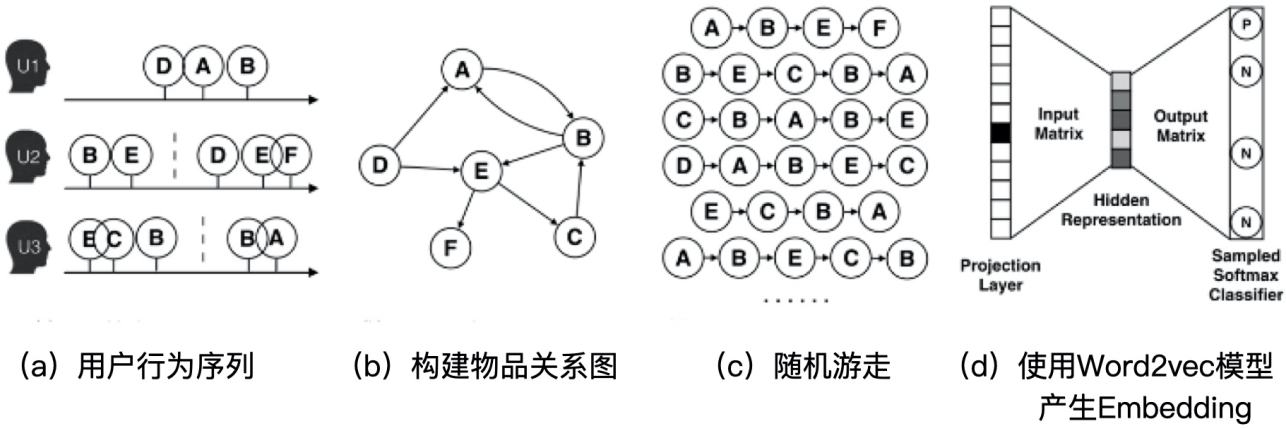

图1 基于随机游走的Graph Embedding算法

我把这类 Graph Embedding 的方法归类为基于随机游走的间接性 Graph Embedding 方法。它其实代表了我们在解决一类技术问题时的思路，**就是面对一个复杂问题时，我们不直接解决它，而是“搭一座桥”，通过这座桥把这个复杂问题转换成一个简单问题，因为对于简单问题，我们有非常丰富的处理手段。这样一来，这个复杂问题也就能简单地解决了。**显然，基于随机游走的 Graph Embedding 方法就是这样一种“搭桥”的解决方案。

但搭桥的过程中难免会损失一些有用的信息，比如用随机游走对图数据进行抽样的时候，虽然我们得到的序列数据中还包含了图结构的信息，但却破坏了这些信息原始的结构。

正因为这样，很多研究者、工程师不满足于这样搭桥的方式，而是希望造一台“推土机”，把这个问题平推过去，直接解决它。GNN 就是这样一种平推解决图结构数据问题的

方法，它直接输入图结构的数据，产生节点的 Embedding 或者推荐结果。当然，不同研究者打造这台推土机的方式各不相同，我们今天要重点介绍的 GraphSAGE，就是其中最著名的一台，也最具参考价值。

GraphSAGE 的主要步骤

下面，我们就来详细讲一讲 GraphSAGE 的细节。GraphSAGE 的全称叫做 Graph Sample and Aggregate，翻译过来叫“图采样和聚集方法”。其实这个名称就很好地解释了它运行的过程，就是先“采样”、再“聚集”。

这时候问题又来了，这里的“采样”还是随机游走的采样吗？要是还通过采样才能得到样本，我们造的还能是“推土机”吗，不就又变成搭桥的方式了吗？别着急，等我讲完 GraphSAGE 的细节，你就明白了。

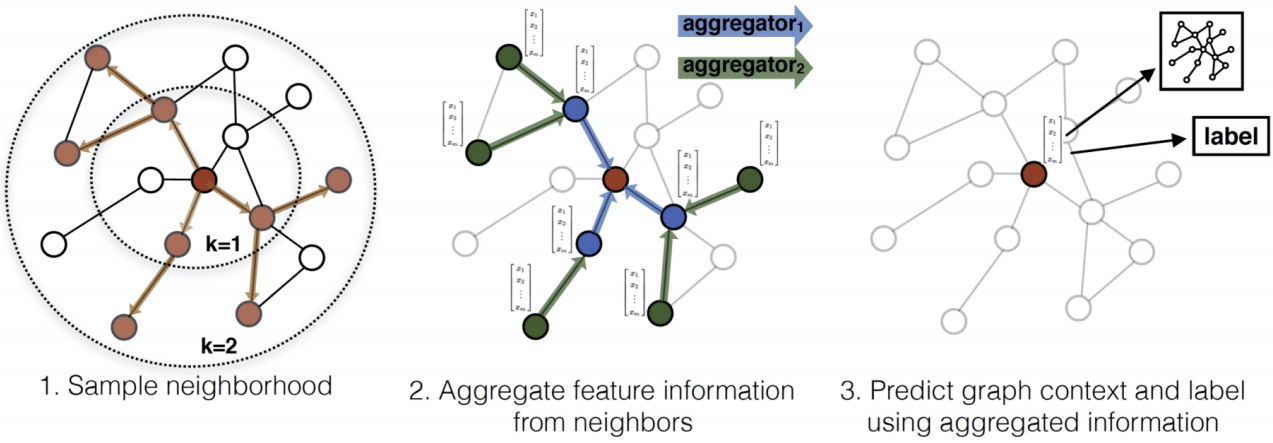

图2 GraphSAGE的主要过程
（出自论文 Inductive Representation Learning on Large Graphs）

GraphSAGE 的过程如上图所示，主要可以分为 3 步：

1. 在整体的图数据上，从某一个中心节点开始采样，得到一个 k 阶的子图，示意图中给出的示例是一个二阶子图；
2. 有了这个二阶子图，我们可以先利用 GNN 把二阶的邻接点聚合成一阶的邻接点（图 1-2 中绿色的部分），再把一阶的邻接点聚合成这个中心节点（图 1-2 中蓝色的部分）；
3. 有了聚合好的这个中心节点的 Embedding，我们就可以去完成一个预测任务，比如这个中心节点的标签是被点击的电影，那我们就可以让这个 GNN 完成一个点击率预估任

务。

这就是 GraphSAGE 的主要步骤，你看了之后可能还是觉得有点抽象。那接下来，我们再结合下图 3 推荐电影的例子，来看一看 GraphSAGE 是怎么工作的。

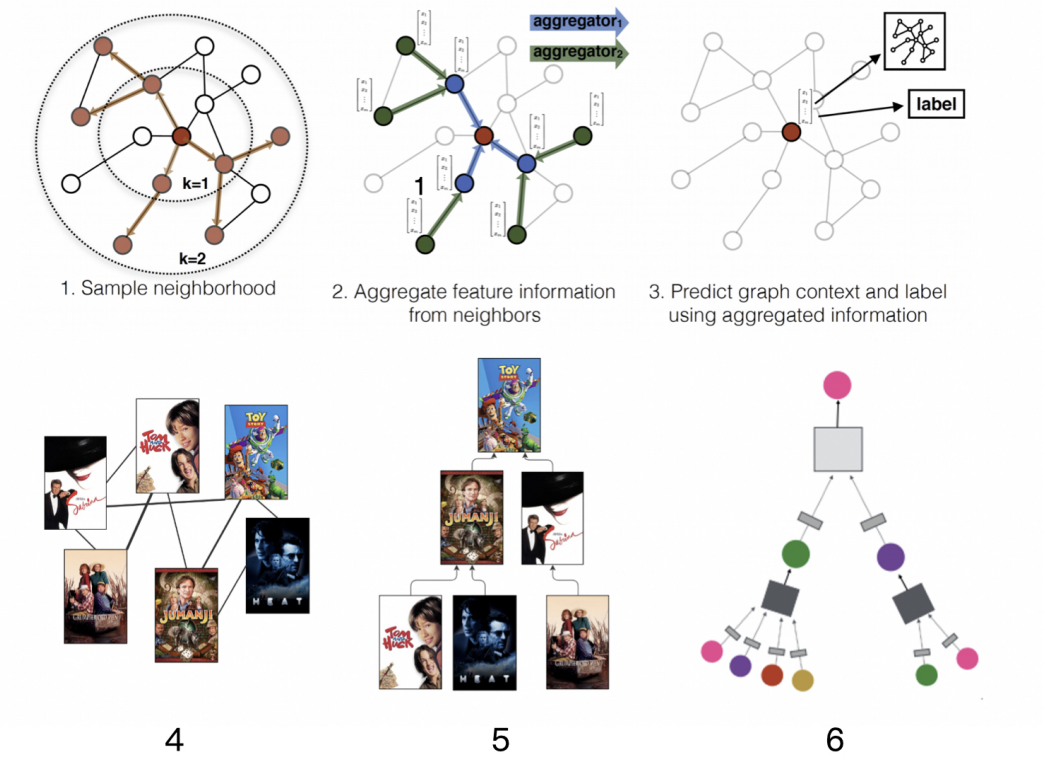

图3 GraphSAGE示例

首先，我们要利用 MovieLens 的数据得到电影间的关系图，这个关系图可以是用用户行为生成（这个方法我们在第 7 节课讲过），它也可以是像生成知识图谱一样来生成，比如，两部电影拥有同一个演员就可以建立一条边，拥有相同的风格也可以建立一条边，规则我们可以自己定。

在这个由电影作为节点的关系图上，我们随机选择一个中心节点。比如，我们选择了玩具总动员（Toy Story）作为中心节点，这时再向外进行二阶的邻接点采样，就能生成一个树形的样本。

经过多次采样之后，我们会拥有一批这样的子图样本。这时，我们就可以把这些样本输入 GNN 中进行训练了。GNN 的结构我会在下一小节详细来讲，这里我们只要清楚，这个 GNN 既可以预测中心节点的标签，比如点击或未点击，也可以单纯训练中心节点的 Embedding 就够了。

总的来说，**GraphSAGE 的主要步骤就是三个“抽样 - 聚合 - 预测”**。

GraphSAGE 的模型结构

现在，我们关注的重点就变成了 GraphSAGE 的模型结构到底怎么样？它到底是怎么把一个 k 阶的子图放到 GNN 中去训练，然后生成中心节点的 Embedding 的呢？接下来，我就结合 GraphSAGE 的模型结构来和你详细讲一讲。

这里，我们还是以二阶的 GraphSAGE 为例，因为超过二阶的结构只是进一步延伸这个模型，没有更多特别的地方，所以我们理解二阶的模型结构就足够了。

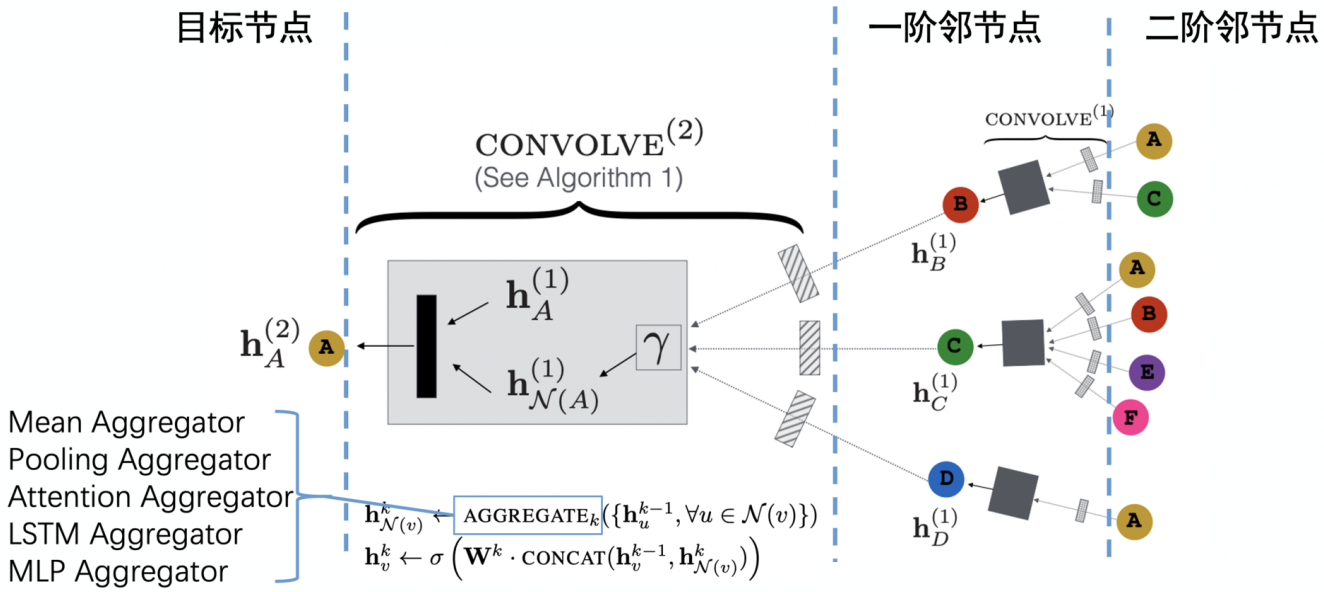

图4 GraphSAGE的模型结构

上图中处理的样本是一个以点 A 为中心节点的二阶子图，从左到右我们可以看到，点 A 的一阶邻接点包括点 B、点 C 和点 D，从点 B、C、D 再扩散一阶，可以看到点 B 的邻接点是点 A 和点 C，点 C 的邻接点是 A、B、E、F，而点 D 的邻接点是点 A。

清楚了样本的结构，我们再从右到左来看一看 GraphSAGE 的训练过程。这个 GNN 的输入是二阶邻接点的 Embedding，二阶邻接点的 Embedding 通过一个叫 CONVOLVE 的操作生成了一阶邻接点的 Embedding，然后一阶邻接点的 Embedding 再通过这个 CONVOLVE 的操作生成了目标中心节点的 Embedding，至此完成了整个训练。

这个过程实现的关键就在于这个叫 CONVOLVE 的操作，那它到底是什么呢？

CONVOLVE 的中文名你肯定不会陌生，就是卷积。但这里的卷积并不是严格意义上的数学卷积运算，而是一个由 Aggregate 操作和 Concat 操作组成的复杂操作。这里，我们要重点关注图 4 中间的部分，它放大了 CONVOLVE 操作的细节。

这个 CONVOLVE 操作是由两个步骤组成的：第一步叫 Aggregate 操作，就是图 4 中 gamma 符号代表的操作，它把点 A 的三个邻接点 Embedding 进行了聚合，生成了一个 Embedding hN(A)；第二步，我们再把 hN(A) 与点 A 上一轮训练中的 Embedding hA 连接起来，然后通过一个全联接层生成点 A 新的 Embedding。

第二步实现起来很简单，但第一步中的 Aggregate 操作到底是什么呢？搞清楚这个，我们就搞清楚了 GraphSAGE 的所有细节。

事实上，Aggregate 操作我们也不陌生，它其实就是把多个 Embedding 聚合成一个 Embedding 的操作，我们在推荐模型篇中也讲过很多次了。比如，我们最开始使用的 Average Pooling，在 DIN 中使用过的 Attention 机制，在序列模型中讲过的基于 GRU 的方法，以及可以把这些 Embedding 聚合起来的 MLP 等等。Aggregate 操作非常多，如果你要问具体用哪个，我还是那句老话，**实践决定最终结构**。

到这里，我们就抽丝剥茧地讲清楚了 GraphSAGE 的每个模型细节。如果你还有疑惑，再回头多看几遍 GraphSAGE 的模型结构图，结合我刚才的讲解，相信不难理解。

GraphSAGE 的预测目标

不过，在讲 GraphSAGE 的主要步骤的时候，我们还留下了一个“小尾巴”没有讲，就是说 GraphSAGE 既可以预测中心节点的标签，比如点击或未点击，又可以单纯地生成中心节点的 Embedding。要知道预测样本标签这个事情是一个典型的有监督学习任务，而生成节点的 Embedding 又是一个无监督学习任务。

那 GraphSAGE 是怎么做到既可以进行有监督学习，又能进行无监督学习的呢？要想让 GraphSAGE 做到这一点，关键就看你怎么设计它的输出层了。

我们先来说说有监督的情况，为了预测中心节点附带的标签，比如这个标签是点击或未点击，我们就需要让 GraphSAGE 的输出层是一个 Logistic Regression 这样的二分类模型，这个输出层的输入，就是我们之前通过 GNN 学到的中心节点 Embedding，输出当然就是预测标签的概率了。这样，GraphSAGE 就可以完成有监督学习的任务了。

而对于无监督学习，那就更简单了。这是因为，我们的输出层就完全可以仿照第 6 节课 Word2vec 输出层的设计，用一个 softmax 当作输出层，预测的是每个点的 ID。这样一来，每个点 ID 对应的 softmax 输出层向量就是这个点的 Embedding，这就和 word2vec

的原理完全一致了。如果你仔细学了 YouTube 的候选集生成模型的话，就会知道这和视频向量的生成方式也是一样的。

GraphSAGE 在 Pinterest 推荐系统中的应用

GraphSAGE 我们讲了这么多，那 Pinterest 到底是怎么在它的推荐系统中应用 GNN 的呢？我这就来讲一讲。

由于 GraphSAGE 是 Pinterest 和斯坦福大学联合提出的，所以 Pinterest 对于 GNN 的应用也是直接在 GraphSAGE 的基础上进行的，只是给这个 GNN 取了个换汤不换药的新名字，PinSAGE。

Pinterest 这个网站的主要功能是为用户提供各种商品的浏览、推荐、收藏的服务，那么所谓的 Pin 这个动作，其实就是你收藏了一个商品到自己的收藏夹。因此，所有的 Pin 操作就连接起了用户、商品和收藏夹，共同构成了一个它们之间的关系图。PinSAGE 就是在这个图上训练并得到每个商品的 Embedding 的。

而 PinSAGE Embedding 的具体应用场景，其实跟我们14 课中实现的功能一样，就是商品的相似推荐。只不过之前业界更多地使用 Item2vec、DeepWalk 这些方法，来生成用于相似推荐的物品 Embedding，在 GNN 流行起来之后，大家就开始尝试使用 GNN 生成的物品 Embedding 进行相似推荐。

那么，PinSAGE 在 Pinterest 场景下的效果到底怎么样呢？Pinterest 给出了一些例子如图 5 所示，我们可以判断一下。

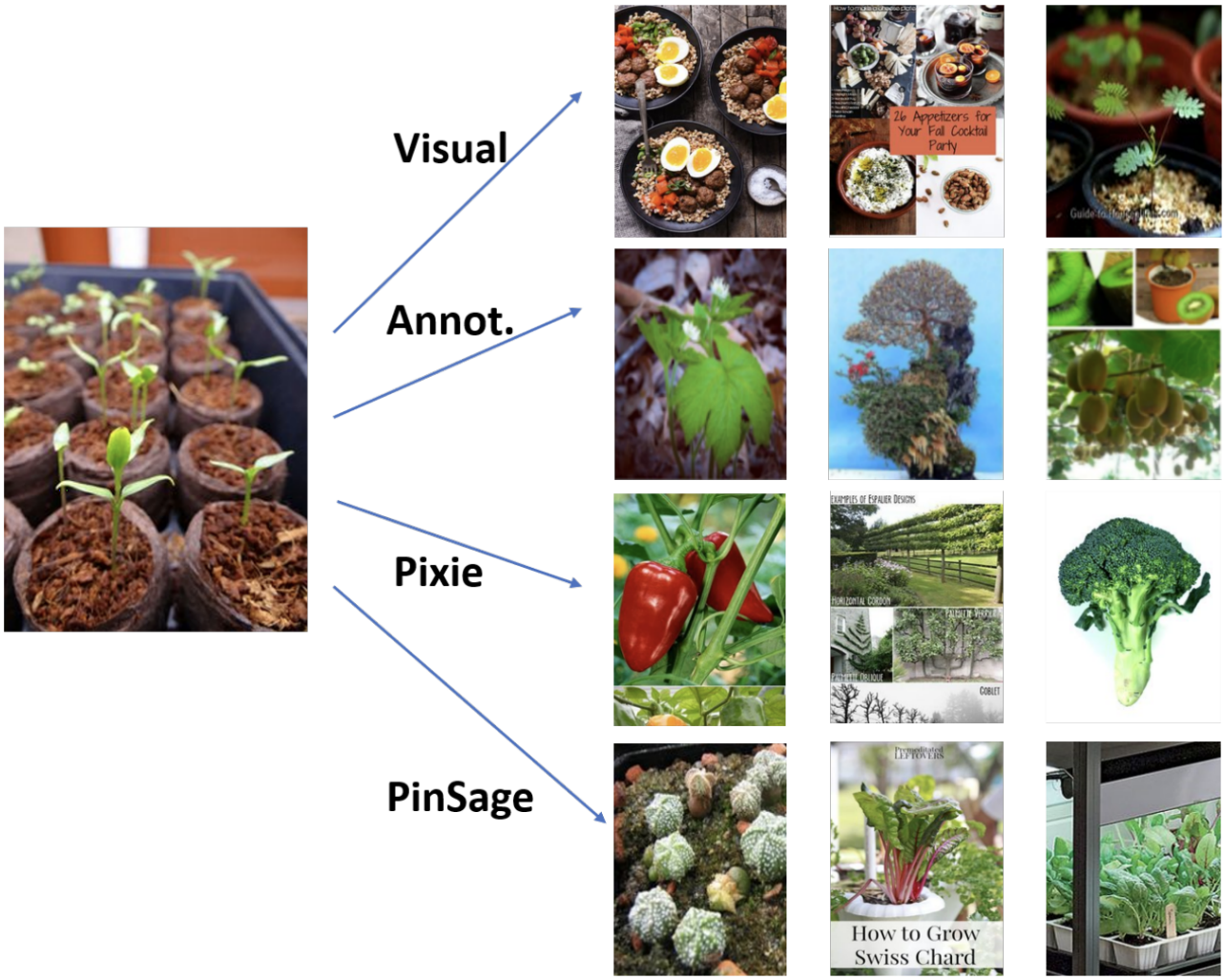

图5 PinSAGE在Pinterest上应用的例子

我们先看图 5 左边的例子，因为它给出的是一个种子发芽的图片，我们就推测它应该是一个卖绿植或者绿植种子的商家。接下来，我们再来判断左边通过四种不同算法找到的相似图片是不是合理。其中，PinSAGE 是 Pinterest 实际用于推荐系统中的算法，其他三个 Visual、Annot、Pixie 都是效果测试中的对比算法。

我们看到通过第一个算法 Visual 找到的图片，虽然看上去和原来的图片比较相似，但前两个图片居然都是食品照片，这显然不相关。第二个算法 Annot 中的树木，以及第三个算法 Pixie 中的辣椒和西兰花，显然都跟绿植种子有很遥远的差距。相比之下，PinSAGE 找到的图片就很合理了，它找到的全都是种子发芽或者培育绿植的图片，这就非常合乎用户的逻辑了。

要知道，在 PinSAGE 应用的构成中，它没有直接分析图片内容，而只是把图片当作一个节点，利用节点和周围节点的关系生成的图片 Embedding。因此，这个例子可以说明，

PinSAGE 某种程度上理解了图片的语义信息，而这些语义信息正是埋藏在 Pinterest 的商品关系图中。可见，PinSAGE 起到了多么神奇的数据挖掘的作用。

小结

这节课，我们讲解了图神经网络的经典方法 GraphSAGE，我们抽丝剥茧地把 GraphSAGE 的细节全部剖开了。关于 GraphSAGE，我们重点要记住它的特点和主要步骤。

首先，GraphSAGE 是目前来说最经典的 GNN 解决方案。因此，它具有 GNN 最显著的特点，那就是它可以直接处理图数据，不需要把图数据转换成更简单的序列数据，再用序列数据 Embedding 方法进行处理。

其次，GraphSAGE 的主要步骤是三步“采样 - 聚合 - 预测”。其中，采样是指在整体图数据上随机确定中心节点，采样 k 阶子图样本。聚合是指利用 GNN 把 k 阶子图样本聚合成中心节点 Embedding。预测是指利用 GNN 做有监督的标签预测或者直接生成节点 Embedding。

在这三步之中，重点在于聚合的 GNN 结构，它使用 CONVOLVE 操作把邻接点 Embedding 聚合起来，跟中心节点上一轮的 Embedding 连接后，利用全连接层生成新的 Embedding。

为了方便你及时回顾，我也把这节课中的重要知识点总结了下面的表格中，你可以看看。

知识点	要点描述
GNN和基于Random Walk的Graph Embedding方法的区别	GNN直接处理图结构数据 Random Walk Graph Embedding需要通过Random Walk的方式把图结构数据转换成序列数据
GraphSAGE方法的主要步骤	抽样–聚合–预测
GraphSAGE模型的主要结构	对于每一阶的邻接点，使用CONVOLVE操作把上一层的邻接点Embedding聚合起来，跟该节点上一轮的Embedding连接后，利用全连接层生成新的Embedding
GraphSAGE的预测目标	可做有监督的标签预测，以及无监督的Embedding生成

极客时间

课后思考

使用 GraphSAGE 是为了生成每个节点的 Embedding，那我们有没有办法在 GraphSAGE 中加入物品的其他特征，如物品的价格、种类等等特征，让最终生成的物品 Embedding 中包含这些物品特征的信息呢？

期待在留言区看到你对 GraphSAGE 的思考，我们下节课见！

提建议

更多学习推荐

© 版权归极客邦科技所有，未经许可不得传播售卖。 页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 28 | 业界经典：YouTube深度学习推荐系统的经典架构长什么样？

精选留言 (4)

写留言

张弛 Conor

2020-12-21

思考题：1）能否在第一阶聚合的时候，就把物品的其他特征拼接起来，作为节点的初始embedding呢？2）也可以在k阶聚合完成后，像wide&deep钟一样，将节点的embedding和物品其他特征拼接后接入全连接层和softmax层得到embedding

展开

作者回复: 很好的回答，第2个方法可能会更直接一些。第一种方法能更好的跟GNN融合在一起。

2

Sicolas Flamel

2020-12-21

我觉得目的导向，以目的来决定要不要更多的特征。资源允许情况下，特征越多越好。喂给机器，由模型来判断哪些特征是重要的，哪些是不重要的。

“这个 CONVOLVE 操作是由两个步骤组成的：第一步叫 Aggregate 操作，就是图 4 中 gamma 符号代表的操作，它把点 A 的三个邻接点 Embedding 进行了聚合，生成了一个 Embedding hN(A)；第二步，我们再把 hN(A) 与点 A 上一轮训练中的 Embedding hA...

展开

作者回复: 完全赞同这样的思考。加在CONVOLVE函数的concat那一步确实会更好的保留特征的影响力。否则我担心经过几轮迭代之后，这些新特征有可能加和不加是一样的，因为整个网络有可能总会收敛到一个稳定的状态。

1

fsc2016

2020-12-21

提问2：既然GraphSAGE是为了直接处理图结构数据生成物品emb，而不需要把图数据转为行为序列。那输入的第k阶物品初始化emb，还是通过转为序列数据进行word2vec得到的。那也就是说GraphSAGE还是避免不了图数据转为序列生成物品emb的过程吗

展开

作者回复: 紧接着上一问，GraphSAGE并不对初始化emb做要求，所以没有必要一定要有word2vec训练好的emb作为初始化emb。

fsc2016

2020-12-21

思考题：第k阶节点emb初始化时怎么做的，如果是基于节点id类特征通过word2vec生成id_emb，那么还可以基于同样方法，生成对应category_emb，brand_emb，price_emb，然后avg pooling。类似前面讲物品冷启动中提到的EGES模型。
提问：感觉如果综合物品所有相关属性生成的物品emb，已经能很好包含了物品所属特征，包括了属性和行为特征，这样生成的emb在输入GraphSAGE得到物品emb，还能得...
展开

作者回复: GraphSAGE原文中的初始化方法其实是随机emb初始化，会随着GraphSAGE的迭代过程逐渐收敛。

但像你说的，这里的灵活性很高，在初始化中以一些简单的emb结果初始化可能会让训练的效率更高，另外加入其他feature当然也是可行的，都是值得尝试的方向。